

Opinnäytetyö (AMK)

tieto- ja viestintätekniikka

2017

Aleksi Nieminen

3D-MALLINNUSPROSESSI PELIKÄYTTÖÖN

Aleksi Nieminen

3D-MALLINNUSPROSESSI PELIKÄYTTÖÖN

Tämän opinnäytetyön tavoitteena oli tutkia 3D-mallinnusprosessin työvaiheita aina suunnittelusta pelimoottoriin viemiseen asti. Prosessin kulusta oli tarkoitus koostaa asiakkaalle optimaalinen, tuotantotehokas ja helposti luettava työnkulunmalli.

Lisäksi toteutettiin asiakkaan peliin esimerkkinä 3D-malli käyttäen tutkittuja työnkulun vaiheita ja soveltaen ne asiakkaan pelille sopiviksi. 3D-mallin tuli olla sellainen, että sitä voitaisiin käyttää pelissä ja siitä kävisi ilmi kaikki 3D-mallintamisen vaiheet.

Työn tuloksena syntyi peliin valmis 3D-malli ja kaavio, josta käy ilmi 3D-mallintamisen eri työvaiheet ja eri työvaiheiden kestot ja tärkeydet. Mallinnettu 3D-malli tulee asiakkaan projektiin sellaisenaan.

ASIASANAT:

3D-mallinnus, mallinnusprosessi, työnkulku, 3ds Max

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information technology | Game technology

2017 | 39

Principal Lecturer Mika Luimula, Adj.Prof

Aleksi Nieminen

3D MODELING PROCESS FOR GAMES

The purpose of this thesis was to investigate all the steps required to complete a 3D-model, from designing to importing the model into desired game engine, and make the most optimal, productive and readable 3D modeling workflow model for the customer.

In addition, a 3D-model is made as an example for a customer's game using the studied workflow steps and applying them to suit the client's needs. The 3D model should be such that it could be used in the game and it would show the workflow and all the steps required for making a 3d model.

The results of the work were a 3D model for the game and a diagram showing the workflow of 3D modeling for games. The Diagram shows all the different stages of 3D-modeling and the duration and importance of each phase. The 3D model was used in the customer's project as such.

KEYWORDS:

3D modeling, workflow, 3ds Max

SISÄLTÖ

KÄYTETYT LYHENTEET JA SANASTO	1-6
1 JOHDANTO	7
2 3D-MALLINNUS YLEISESTI	8
2.1 Mallinnusmenetelmät	9
2.2 Korkea- ja matalapolygoniset mallit	12
2.3 Työkalut ja modifierit	13
3 3D-MALLINTAMISEEN KÄYTETTÄVÄT OHJELMISTOT	17
3.1 3ds Max	17
3.2 Blender	19
3.3 Muita 3D-mallinnus ohjelmistoja	20
3.4 Yhteenveto	21
4 3D-MALLINNUSPROSESSI	22
4.1 Suunnittelu vaihe	22
4.2 Mallinnus	23
4.3 Optimointi ja viimeistelyvaihe	24
4.4 Riggaus ja Animointi	25
4.5 UV-Kartoitus	25
4.6 Beikkaus	27
4.7 Exporttaus	28
4.8 Teksturointi	28
4.9 Importtaus	30
5 MALLINNUSPROSESSIN HYÖDYNTÄMINEN PELIPROJEKTIIN	31
5.1 Lyhyesti peliprojektista	31
5.2 Suunnitteluvaihe	31
5.3 Mallinnusvaihe	32
5.4 Optimointi	33
5.5 UV Mappaus	33
5.6 Exporttaus	34
5.7 Teksturointi	34
5.8 Mallin vienti pelimoottoriin	35

6 YHTEENVETO

37

LÄHTEET

38

KUVAT

Kuva 1. X-, Y- ja Z-koordinaattiakselit.	8
Kuva 2. polygonin rakenne (Autodesk, 2015.)	8
Kuva 3. Primitiivisiä malleja ja 3ds Maxin tukemat primitiiviset muodot.	9
Kuva 4. NURBS-tekniikalla toteutettu malli ja 3ds Maxin NURBS-ominaisuudet.	10
Kuva 5. Esimerkki sculptatusta 3D-mallista (CGMeetup, 2015.)	11
Kuva 6. Extrude-työkalu 3ds Maxissa.	13
Kuva 7. Chamfer-työkalu 3ds Maxissa.	13
Kuva 8. Bridge Edges-työkalu 3ds Maxissa.	14
Kuva 9. Connect Edges-työkalu 3ds Maxissa.	14
Kuva 10. Weld Vertices -työkalun alku- ja lopputilanne 3ds Maxissa.	15
Kuva 11. Esimerkki 3D-mallien tasoitusrhymistä.	15
Kuva 12. Vasemmalla normaali laatikko ja oikealla kyseinen laatikko TurboSmoothattuna.	16
Kuva 13. Yleiskuva 3ds Maxin käyttöliittymästä.	18
Kuva 14. Yleiskuva Blenderin käyttöliittymästä.	19
Kuva 15. Esimerkki hyvästä ja huonosta kasvojen topologiasta. (Thilakanathan Studios, 2016.)	23
Kuva 16. Esimerkkejä yhden objektin LOD-malleista (Polygon Reducer, 2008.)	24
Kuva 17. Taso- ja sylinterimappaukset	26
Kuva 18. Pallo- ja kutistusmappaukset.	26
Kuva 19. Laatikkomappaus ja laatikon unwrapattu uv-kartta.	27
Kuva 20. Videokameran referenssikuvat (Sony.)	32
Kuva 21. Valmis 3D-malli ilman tekstuureja ja valmiiksi määritellyillä materiaalipaikoilla.	33
Kuva 22. Valmis videokameran 3D-malli teksturointuna ja pelimoottoriin vietyinä.	36

KÄYTETYT LYHENTEET JA SANASTO

Assetti	Peliin tuleva "asia" mm. 3D-malli, äänitehoste, tekstuuri
Blueprint	Unreal Enginen käyttämä visuaalinen skriptausjärjestelmä.
Edge	Kahden verteksin välinen reuna.
Frame	Kuva (frames per second = kuvaa sekunnissa)
Keyframe	Määrittää animation alkamis- ja loppumispisteet.
Modifier	Määrite
Node	Blueprinteissa käytettävä objekti.
Polygoni	Verteksien väliin jäävä alue.
Renderöinti	Kuvan luominen 3D-mallista.
Riggaus	Luiden luominen animoitavalle 3D-mallille.
Skinnaus	Luotujen luiden sitominen animoitavaan 3D-malliin.
Staatinen meshi	Peliin tuleva 3D-malli, jota ei animoida.
Verteksi	Piste 3D-maailmassa.
Whiteboxing	Primitiivimuodoilla luotu luonnos pelialueen objekteista.

1 JOHDANTO

Opinnäytetyön alkupuolella käsitellään 3D-mallinnuksen teoriaa. Hankittua tietoa analysoidaan ja pohditaan sitä, miten saatua tietoa voidaan soveltaa vastaamaan toimeksiannon tarpeita. Opinnäytetyössä vertaillaan tarkemmin kahta eri 3D-mallinnusohjelmistoa: Blenderia ja 3ds Maxia. Työssä kerrotaan myös kolmesta muusta 3D-mallinnusohjelmasta.

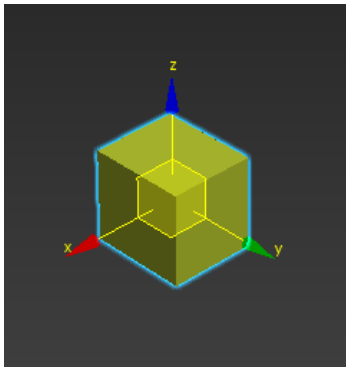
Opinnäytetyön toimeksiantajana toimii turkulainen peliyhtiö Morrow Games, ja opinnäytetyö toteutetaan osana Morrow Gamesin peliprojektia. Opinnäytetyössä tutkitaan 3D-mallinnuksen eri osa-alueita ja käydään läpi 3D-mallin tekemisen eri vaiheet läpi. Sen lisäksi kuvataan, miten syntynyt idea toteutetaan valmiiseen peliin. Esimerkkimallina toimii kannettava videokamera, jonka on tarkoitus valmistuttuaan olla osa Morrow Gamesin peliprojektia. Videokamera valikoitui esimerkkimalliksi, koska sitä mallintaessani käyn lähes työvaihemallin kaikki työvaiheet läpi.

Projektin pelimoottoriksi valikoitui Epic Gamesin kehittämä Unreal Engine 4, koska se on ollut eniten asiakkaan käytössä ollut pelimoottori. Esimerkkimalli mallinnettiin Autodeskin kehittämällä 3ds Max 2016 -ohjelmistolla, koska ohjelmisto on helppokäyttöisempi kuin muut vastaavat. Teksturointiohjelmistona käytettiin Allegorithmicin kehittämää Substance Painteria, koska kyseinen ohjelmisto on suoraan yhteensopiva peliprojektissa käytetyn pelimoottorin kanssa.

Morrow Games on perustettu 2013 ja alussa se keskittyi mobiiliapplikaatioiden tekemiseen. Yritys on kehittänyt pelejä vuodesta 2015 ja nykyään yrityksen pääpaino on Virtual Reality -pelien kehityksessä. Yritys on julkaissut yhden pelin HTC Vivelle Steamiin joulukuussa 2016.

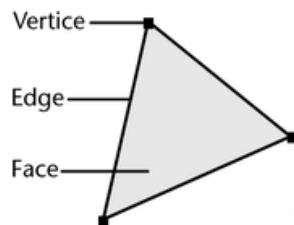
2 3D-MALLINNUS YLEISESTI

3D-mallinnus on prosessi, jonka tarkoituksena on kehittää matemaattinen esitys minkä tahansa kohteen kolmiulotteisesta pinnasta. Kohde voi olla joko eloton tai elävä. Valmista tuotetta kutsutaan 3D-malliksi, joka voidaan näyttää 3D-renderöitynä kaksiulotteisena kuvana tai tietokoneilla tehdyissä simulaatioissa. 3D-mallin pystyy myös fyysisesti luomaan käyttämällä 3D-tulostuslaitteita. Mallinnustapa on noussut yhä suurempaan suosioon viime vuosikymmeninä eri medioissa, kuten peleissä, elokuvissa ja mainoksissa. 3D-malleja käytetään myös lääketieteessä ja teollisuudessa. 3D-mallit piirretään x-, y- ja z-koordinaattiakselien avulla. (wiseGEEK, 2017.)



Kuva 1. X-, Y- ja Z-koordinaattiakselit.

3D-malli koostuu polygoneista ja polygoni muodostuu pisteistä (verteksi), reunoista (edge) ja pinnasta (face) (Kuva 2.). Yleisesti ottaen polygoni on mikä tahansa suoraviivainen muoto, joka on määritelty kolmiulotteisilla pisteillä ja ne liittäville reunoilla. Reunojen ja pisteiden väliin jäävää aluetta kutsutaan faceksi. (Autodesk, 2015.)



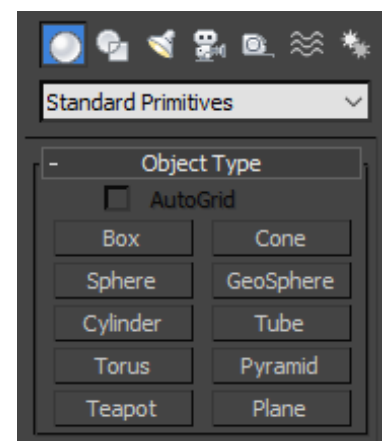
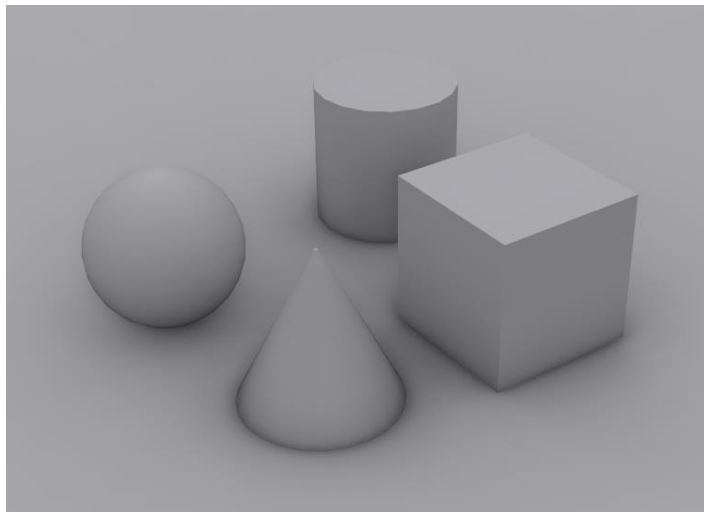
Kuva 2. polygonin rakenne (Autodesk, 2015.)

Elokuviin ja muihin medioihin tehtävä 3D-mallintaminen eroaa peleihin tehtävistä mallinnuksista, koska pelit ovat interaktiivisia. Vaikka peliä suunnittelisi kuinka paljon, on lähes mahdotonta varautua kaikkiin mahdollisiin tilanteisiin, joihin pelaaja voi pelissä päätyä. Peleissä siis käytetään pelimoottoria ja reaaliaikaista renderöintiä, kun taas elokuvissa renderöinti tulee tehdä ennen kuin katsoja saa elokuvan nähtäväkseen. Esimerkiksi animaatioelokuvissa yhden kuvan renderöimiseen voi mennä useita tunteja. Koska pelien 3D-mallit renderöidään reaaliajassa, mallin polygonimäärään ja optimointiin tulee kiinnittää erityistä huomiota.

2.1 Mallinnusmenetelmät

Primitiivinen mallinnus

On olemassa tyypillisesti neljä mallintamismenetelmää. Ensimmäinen ja perinteisin tapa tunnetaan primitiivisenä mallintamismenetelmänä. Tämä on yksinkertaisin tapa mallintaa 3D-esineitä, ja siinä käytetään geometrisiä perusmuotoja, kuten palloja, kartioita, kuutioita ja sylintereitä. Muodot ovat yleensä matemaattisesti määriteltyjä, mikä helpottaa työskentelyä. Primitiivistä mallinnusta käytetään pääasiassa teknisten sovellusten 3D-malleihin ja placeholder-grafiikkana peleihin. (wiseGEEK, 2017.) Esimerkiksi 3ds Maxissa on jo valmiina primitiivisiä muotoja. (Autodesk, 2017.)



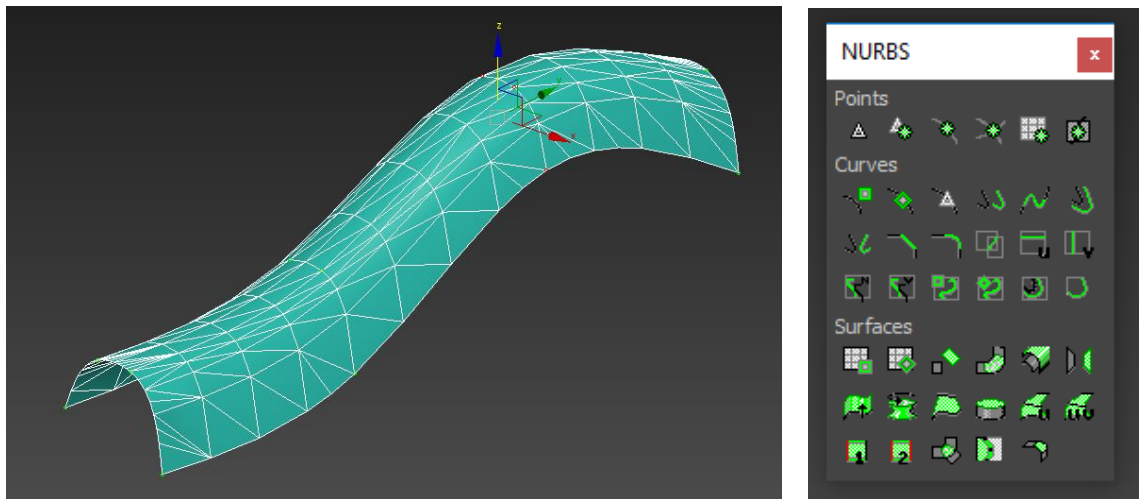
Kuva 3. Primitiivisiä malleja ja 3ds Maxin tukemat primitiiviset muodot.

Polygonaalinen mallinnus

Polygonaalinen mallinnus on primitiivistä mallintamista hieman edistyneempi tapa mallintaa. Polygonaalisessa mallinnuksessa usein primitiiviseen malliin lähdetään lisäämään polygoneja erilaisten modifierien avulla. Polygonaaliset mallit ovat erittäin joustavia ja ne pystytään renderöimään tietokoneella erittäin nopeasti. Polygonaalista mallinnustapaa kutsutaan myös laatikkomallinnukseksi, koska useimmin mallinnus aloitetaan pelkästä laatikosta. Polygonaalisella mallinnuksessa ei kuitenkaan pystytä luomaan tarkkaa kaarevaa pintaa. (Autodesk, 2017.) Tämä rajoittaa mallinnustekniikan käyttökelpoisuutta joissakin sovelluksissa. Polygonaalinen mallinnus on yleisin käytetty mallinnusmenetelmä.

NURBS

NURBS eli Non-uniform rational B-Spline modelling on yksi parhaista tavoista käyttäjälle luoda 3D-malli, jossa on kaarevat pinnat. Toisinkuin polygonaalinen mallinnustekniikka, joka vain mallintaa kaarevia pintoja likimääräisesti käyttäen lukuisia polygoneja, NURBS-mallinnus taivuttaa tilaa. NURBS-mallinusta käytetään suunnitteluprosessissa tarkan ja täsmällisen käyrän kuvauksen määrittämiseen. NURBS-käyriä käytetään esimerkiksi aerodynamiikassa, auton koreissa ja muissa malleissa, jossa pintojen sujuva kaarevuus on tärkeää. (Rhinoceros, 2017)



Kuva 4. NURBS-tekniikalla toteutettu malli ja 3ds Maxin NURBS-ominaisuudet.

Skulptaus

Skulptaus (englanniksi sculpting, veistäminen) tarkoittaa sellaisten ohjelmistojen käyttöä, jotka tarjoavat työkaluja digitaalisen objektin työntämiseen, pinoamiseen, vetämiseen, tarttumiseen tai liikkumiseen samalla tavoin, kuin objekti olisi tehty jostain oikean maailman aineesta, kuten esimerkiksi savesta. Skulptauksella saadaan aikaan tarkkoja yksityiskohtia, jotka olisivat lähes mahdottomia mallintaa muilla mallinnusmenetelmillä. Skulptaustekniikka mahdollistaa fotorealististen mallien tuottamisen. Skulptaustekniikalla tuotettuja malleja käytetään laajalti lähinnä elokuvissa, taiteessa, fotorealistisissa kuvissa ja 3D-tulostuksessa. Skulptausohjelmistoja ovat muun muassa Zbrush ja Autodeskin Mudbox. (Papageorgiou, 2017.) Myös Blenderillä pystyy skulptaamaan. (Blender, 2017.) Skulptausta käytetään etenkin orgaanisten mallien mallinnuksessa.



Kuva 5. Esimerkki sculpatusta 3D-mallista (CGMeetup, 2015.)

2.2 Korkea- ja matalapolygoniset mallit

Korkeapolygonisen (high poly) mallin tarkoituksena on saavuttaa mahdollisimman realistinen lopputulos. Korkeapolygonisten mallien ongelmana kuitenkin on verteksien valtava määrä, ja niiden hallitseminen. Tämä saa aikaan sen, että korkeapolygoniset mallit ovat suorituskyvyn kannalta hieman ongelmallisia. Koska mallia ei tarvitse renderöidä reaaliajassa, korkeapolygonisen mallin polygonimäärä voi olla useita miljoonia. korkeapolygonisia malleja käytetään, kun halutaan taata mahdollisimman korkea laatu mallille. (Autodesk 2017.)

Matalapolygonisia (low poly) malleja käytetään useimmiten peleissä tai korkeapolygonisten mallien riggauksessa ja animoinnissa. Matalapolygonisten mallien polygonimäärä riippuu kohdealustasta. Peleissä käytetään usein vain matalapolygonisia malleja. (Autodesk 2017.)

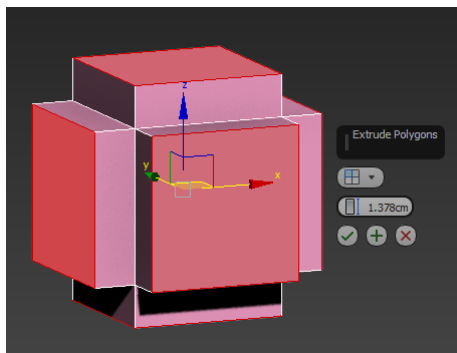
Se mitä pidetään korkeapolygonisena ja matalapolygonisena vaihtelee ajan kuluessa ja tietokoneiden kehittyessä. Yleisesti matalapolygonisena mallina pidetään mallia, joka voidaan renderöidä reaaliajassa kohdealustalle. Tämä tarkoittaa, että mobiililaitteille matalapolygoninen malli on eri kuin pöytätietokoneelle.

Korkeapolygonisesta mallista saa matalapolygonisen mallin muuttamalla mallin topologiaa ja yksinkertaisesti vähentämällä polygoneja, mutta silti pitäen mallin muodon. korkeapolygonisen mallin yksityiskohdat voidaan myös beikata tekstuurikartoiksi, jolloin matalapolygoninen malli saadaan näyttämään korkeapolygoniselta, kuitenkin säästäten polygoneissa ja suorituskyvyssä. Matalapolygonisesta mallista taas saa korkeapolygonisen mallin esimerkiksi sculptaamalla.

2.3 Työkalut ja modifierit

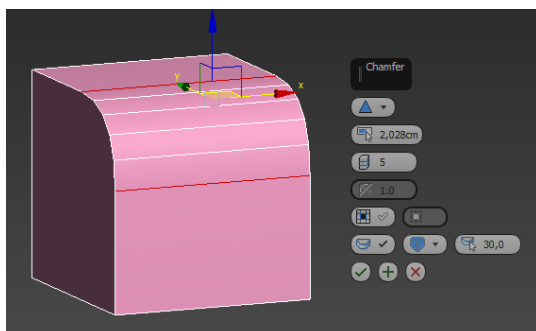
Kaikissa mallinnusohjelmissa on hyvin saman tyyppisiä työkaluja ja modifiereita 3D-mallin mallintamista varten. Tässä kappaleessa käydään läpi muutamia 3ds Maxin yleisimmin käytettyjä työkaluja ja modifiereita.

Extrude-työkalulla mallin valittuja polygoneja työnnetään, joko sisälle tai ulospäin objektista, säilyttäen kuitenkin objektin alkuperäisen geometrian ja luoden ”sillan” alkuperäisen ja extrudatun kohdan välille. Extrude-työkalu toimii myös mallin reunoihin ja yksittäisiin vertekseihin. (Autodesk, 2017.)



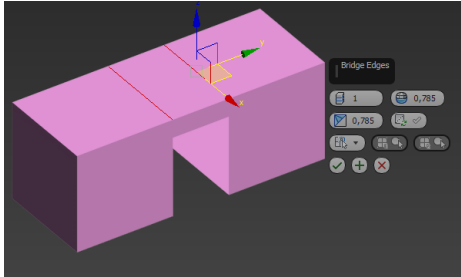
Kuva 6. Extrude-työkalu 3ds Maxissa.

Chamfer-työkalun avulla 3D-mallia pystyy elävöittämään mallin reunoja pyöristämällä. Chamfer pyöristää monistamalla valittuja reunoja käyttäjän syöttämien arvojen perusteella. Chamfer työkalusta löytyy kaksi asetusta: Standard Chamfer ja Quad Chamfer. Quad Chamfer luo uusia polygoneja jokaisen pyöristetyn kulman ympärille. (Autodesk, 2017.)



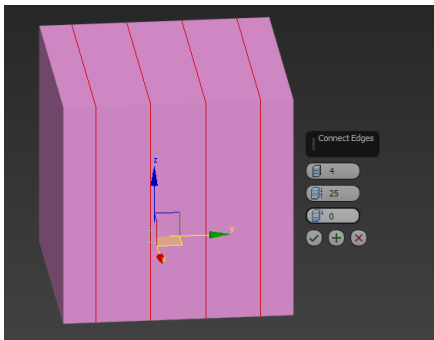
Kuva 7. Chamfer-työkalu 3ds Maxissa.

Bridge-työkalu muodostaa nimensä mukaisesti sillan kahden polygonin, kahden avoimen edge:n tai kahden avoimen borderin välille. Bridge-työkalun avulla voi yhdistää kaksi objektia. (Autodesk, 2017.)



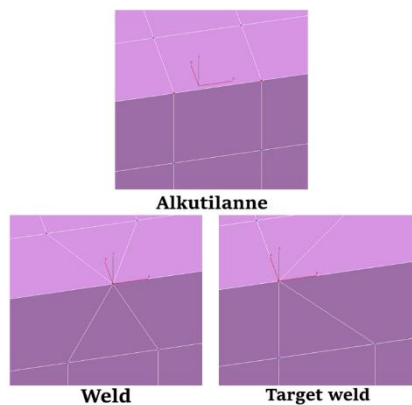
Kuva 8. Bridge Edges-työkalu 3ds Maxissa.

Connect -työkalu luo haluttujen edgejen välille uusia edgejä. Työkalun avulla voidaan nostaa mallin polygonimäärää, ja helposti saada mallista halutun muotoinen. Connect -työkalulla voi määrittellä uusien reunojen määrän sekä reunojen paikan, joko toisistaan tai yleisestä sijainnista. (Autodesk, 2017.)



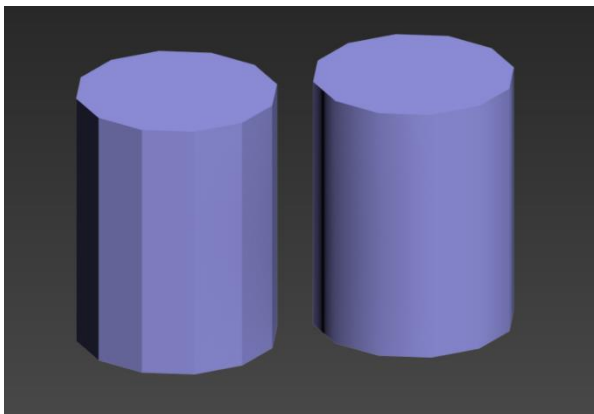
Kuva 9. Connect Edges-työkalu 3ds Maxissa.

Weld Vertices toiminnon avulla voidaan kaksi tai useampia verteksejä yhdistää eli "hitsaa" yhdeksi verteksiksi. Toiminnolla voi määrittellä maksimietäisyyden jonka sisällä olevat verteksit hitsataan. Kynnysarvon ulkopuolelle jääviä verteksejä tai edgejä ei hitsata. Työkalu myös näyttää verteksien lukumäärän ennen ja jälkeen hitsauksen. Verteksien hitsaus on hyvä tapa optimoida valmista mallia. Target Weld - toiminnon avulla kaksi vertekstiä voidaan yhdistää toisen verteksin paikalle, kun taas weld toiminnolla hitsauskohta tulee verteksien väliin jäävän alueen keskelle. (Autodesk, 2017.)



Kuva 10. Weld Vertices -työkalun alku- ja lopputilanne 3ds Maxissa.

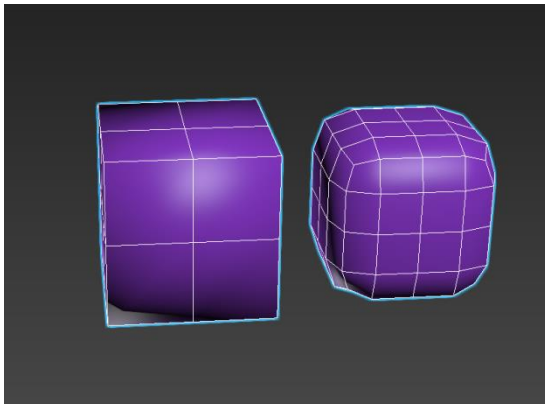
Tasointusryhmät (englanniksi Smoothing groups) määrittävät, renderöidäänkö mallin pinta terävinä reunoina vai sileinä pintoina. Tasointusryhmät ovat hyödyllisiä sellaisten muotojen kuvaamiseen, joissa jotkut polygonit ovat sujuvasti yhteydessä viereisiin polygoneihin ja toiset polygonit taas eivät ole. (Autodesk, 2017.) Esimerkiksi sylinterin muotoisessa objektissa, kaikki polygonit ovat toisiinsa yhteydessä sujuvasti, paitsi päätykappaleet. Päätykappaleet ovat omassa tasointusryhmässään ja loput omassaan. Kuvassa 11. on vierekkäin kaksi täysin samanlaista objektia, joista oikeanpuoleisessa on käytössä tasointusryhmät ja vasemmanpuoleisessa tasointusryhmät ovat pois päältä.



Kuva 11. Esimerkki 3D-mallien tasointusryhmistä.

TurboSmooth -työkalun avulla voidaan jakaa geometriaa interpoloimalla objektin pintojen kaltevuutta polygonien kulmissa ja reunoilla. TurboSmooth vaikuttaa objektiin pyöristämällä kulmien ja reunojen yli ikään kuin ne olisivat hiottu pyöreiksi. TurboSmoothin parametrejä säätelemällä voidaan määrittää objektin iteraatioiden ja

facejen määrä. (Autodesk, 2017.) Kuvassa 12. on vasemmalla normaali laatikko ja oikealla kyseinen laatikko TurboSmooth –toimintoa käyttäen.



Kuva 12. Esimerkki TurboSmooth -toiminnosta

3 3D-MALLINTAMISEEN KÄYTETTÄVÄT OHJELMISTOT

3D-mallinnukseen on tarjolla useita eri ohjelmistoja, joista käyttäjä voi valita itselleen omaan tarkoitukseensa sopivimman. Tässä kappaleessa kerrotaan ensin kattavasti 3ds Maxista ja Blenderistä. Sen jälkeen kuvataan lyhyesti muita saatavilla olevia ohjelmistoja antamaan kattavamman kuvan saatavilla olevista 3D-mallinnusohjelmistoista. Lopuksi vertaillaan 3ds Maxia ja Blenderiä keskenään.

3.1 3ds Max

3ds Max (Aikaisemmin 3D Studio, myöhemmin 3D Studio Max) on yksi suosituimmista ja ammattilaisten keskuudessa eniten käytetyimmistä 3D-mallinnusohjelmistoista. Sitä kehittää ja julkaisee Autodesk. 3ds Max on suosittu työkalu pelinkehittäjien, TV:n mainosstudioiden ja arkkitehtuuriin liittyvää visualisaatiota tekevien studioiden keskuudessa. Ensimmäinen versio 3ds Maxista ilmestyi jo vuonna 1990 ja viimeisin versio Autodesk 3ds Max 2018 huhtikuussa 2017. (Time & Trends Academy, 2016.)

Vuonna 1988 Gary Yost perusti Yost Groupin ja aloitti yhteistyön Autodeskin kanssa. Samaan aikaan tiimi aloitti uuden mallinnus- ja renderöinti sovelluksen kehittämisen. Projektin nimeksi tuli THUD. Sovelluksessa oli neljä moduulia: Shaper, Loftter, Editor ja Material Editor, johtuen DOS:n 640k muistin rajasta. Dan Silvan kanssa aloitetun yhteistyön myötä keyframing -toiminto otettiin käyttöön, ja siitä tuli viides ja viimeinen moduuli, 3D Studio. THUD muuttui 3D-animaatio-ohjelmaksi, jonka johdosta Autodesk suostui julkaisemaan 3D Studion ja poistamaan 640k muistin rajan, jolloin kaikki moduulit toimivat yhdessä. (Time & Trends Academy, 2016.)

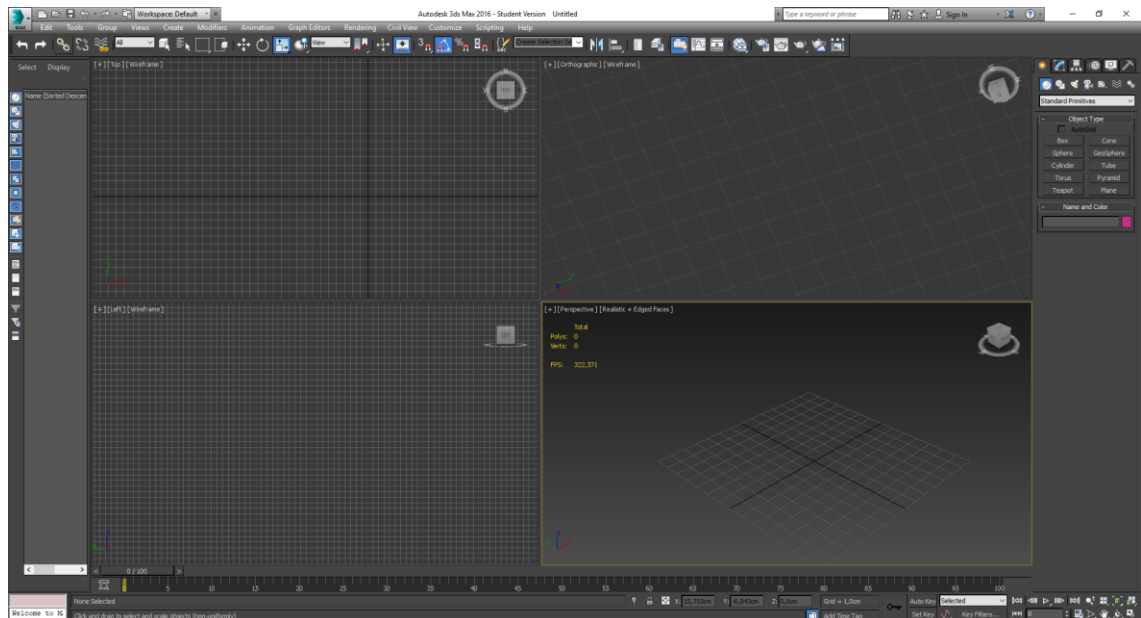
Vuonna 1990 Autodesk julkaisi 3D Studion, joka oli ensimmäinen edullinen ja integroitu 3D-mallinnus, renderöinti ja animaatio-ohjelmisto PC:lle. Tuote julkaistiin halpaan 3495 Yhdysvaltain dollarin hintaan. Tätä aiemmin 3D-mallinnus ohjelmistot olivat kalliita ja ne toimivat vain kalliilla SGI-koneilla. Hinta ei ole noussut 3D Studion julkaisun jälkeen olleenkaan. Ensimmäisessä 3D Studiossa oli toimintoina muun muassa kyky piirtää spline-muotoja ja geometrisiä primitiivejä ja keyframing yhdistettynä perus meshin muokkaukseen. (Time & Trends Academy, 2016.)

Vuosina 1992–1994 ilmestyi 3D Studio versiot 2,3 ja 4. Uusia ominaisuuksia näissä versioissa olivat muun muassa GUI:n kustomointi ja Material Editorin sekä Rendererin perinpohjainen kunnostus. Vuonna 1996 ilmestyi ensimmäinen versio 3D Studio Max -nimellä Windows NT:lle. (Time & Trends Academy, 2016.)

3ds Maxin julkaisuissa on siirrytty niin sanottuun rolling release -päivätyöfilosofian puolelle. Tämä tarkoittaa sitä, että yksittäisiä versioita ei pysty enää ostamaan, vaan rahoitusmalli on siirtynyt kuukausi- tai vuosimaksulla toimivaan subscriptioniin. 3ds Maxin korkean hinnan takia, ohjelmasta on saatavissa myös ilmainen opiskelijaversio, jonka tarkoituksena on tuoda opiskelijoille mahdollisuus ohjelman opettelemiseen.

3ds Max tukee lähes kaikkia 3D-mallinnuksen vaiheita. 3ds Maxissa pystyy esimerkiksi animoimaan, riggaamaan, skinnaamaan, luomaan yksinkertaisia materiaaleja, beikkaamaan, renderöimään ja unwrappaamaan UV mappeja. (Autodesk, 2017.)

Yksi 3ds Maxin ominaisuuksista on MAXScript. MAXScript on 3ds Maxin sisäänrakennettu scriptauskieli. MAXScriptin avulla käyttäjällä on mahdollisuus scriptata kaikkia 3ds Maxin ominaisuuksia, kuten mallinnusta, animaatiota, materiaaleja, renderausta ja niin edelleen. Käyttäjä pystyy myös luomaan omia importtausta ja exporttausta toimintoja. (Autodesk, 2017.)



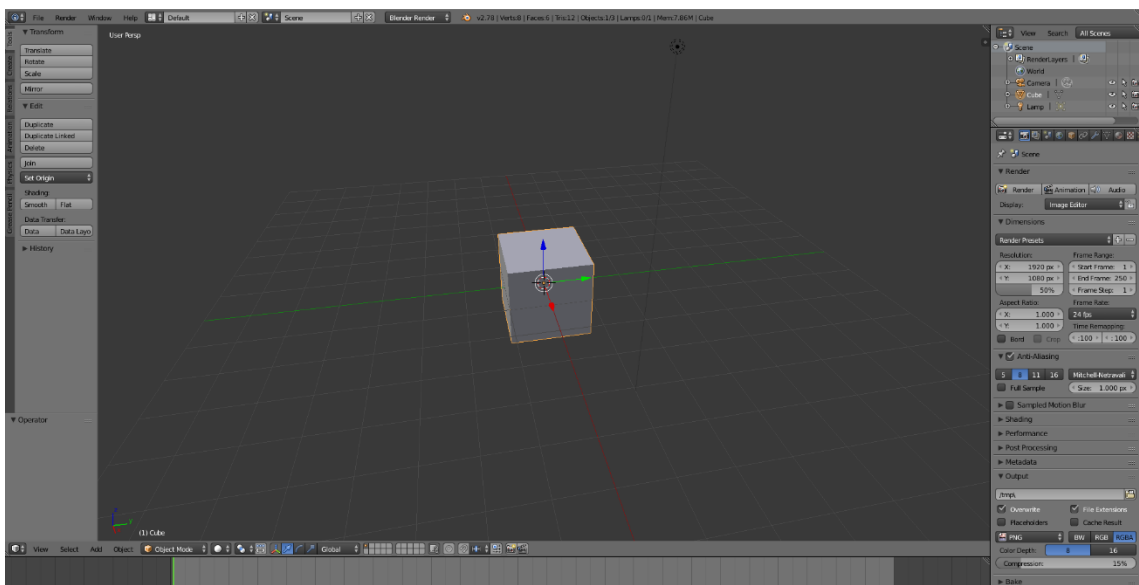
Kuva 13. Yleiskuva 3ds Maxin käyttöliittymästä.

3.2 Blender

Blender on avoimen lähdekoodin 3D-mallinnus ohjelmisto. Blender toimii Windowsin lisäksi Linuxilla ja Macilla. Vuonna 1995 hollantilainen Ton Roosendaal aloitti vanhan yrityksensä 3D-työkalun uudelleenkirjoituksen aloittaen aivan alusta. Tämä löi alkutahdit 3D-työkalulle, joka nykyisin tunnetaan nimellä Blender. (Blender Foundation, 2017.)

Vuonna 1998 Roosendaal perusti yrityksen nimeltä Not a Number (NaN), tarkoituksenaan markkinoida ja kehittää Blenderiä. Yrityksen liiketoimintamalli koski kaupallisia tuotteita ja palveluita Blenderin ympärillä, ja vuonna 2000 yritys rahoitti kasvuaan useiden sijoitusyhtiöiden kautta. Blenderin tavoitteena oli luoda ilmainen työkalu vuorovaikutteisiin 3D-sisältöihin ja kaupallisiin versioihin ohjelmiston jakeluun ja julkaisemiseen. Kuitenkin vuonna 2002 sijoittajat päättivät lopettaa yrityksen tukemisen alhaisen myynnin ja vaikean taloudellisen tilanteen vuoksi. (Blender Foundation, 2017)

Käyttäjäyhteisön innostunut tuki kuitenkin sai aikaan sen, että vuonna 2002 Roosendaal perusti voittoa tavoittelemattoman Blender Foundationin. Blender –säätiön tarkoituksena oli löytää keino, jolla kehittää ja edistää Blenderiä yhteisöpohjaisena avoimen lähdekoodin hankkeena. Tämän seurauksena ”Free Blender” –kampanja käynnistettiin, ja se keräsi 100 000 euroa seitsemässä viikossa. Vuonna 2002 Blender julkaistiin GNU General Public License –ehtojen mukaisesti. (Blender Foundation, 2017.)



Kuva 14. Yleiskuva Blenderin käyttöliittymästä.

Blender tukee kaikkia 3D-mallintamisen pipelineja – mallintamista, riggausta, animointia, simulointia, renderöintiä, kompositointia ja liikkeen seuranta. Blenderin ominaisuuksiin kuuluu myös videon editointi ja pelien luominen. Käyttäjät voivat myös käyttää Blenderin ohjelmointirajapintaa, kustomoidakseen ja kirjoittaakseen uusia ominaisuuksia blenderiin. Blender tukee Linux-, Windows- ja Macintosh-käyttöjärjestelmiä. Blenderin käyttöliittymä käyttää OpenGL:ää, joka on laitteistoriippumaton ohjelmointirajapinta. (Blender Foundation, 2017.)

3.3 Muita 3D-mallinnus ohjelmistoja

3D-mallinnusohjelmistoja on useita erilaisia ja varmasti kaikille löytyy omia tarpeita vastaava ohjelma. Muita 3D-mallinnusohjelmistoja ovat muun muassa: Cinema 4D, Maya, Modo. Tässä kappaleessa käydään ohjelmistot läpi lyhyesti.

Cinema 4D on animointiin ja 3D-mallinnukseen tarkoitettu ohjelma. Sen kehittäjä on Maxon. Cinema 4D sai alkunsa vuonna 1990 kun saksalaiset veljekset Christian ja Philip Losch osallistuivat säteenseurantaohjelmallaan Kickstart-lehden kuukausittaiseen ohjelmointi kilpailuun, jonka he voittivat. Seuraavana vuonna FastRay ilmestyi amigalle. FastRay vaihtoi nimensä Cinema 4D:ksi vuonna 1993. Viimeisin versio Cinema 4D:stä on Cinema 4D R18, ja se on ilmestynyt vuonna 2016. Cinema 4D:tä käytetään laajalti elokuvateollisuudessa. Sitä on käytetty suurissa elokuvaprojekteissa, joita ovat esimerkiksi The Martian ja Inception. (Maxon, 2017.)

Maya on Autodeskin julkaisema ja kehittämä ammattikäyttöön suunnattu 3D-mallinnus ohjelmisto. Sitä käytetään interaktiivisten 3D-sovellusten, kuten videopelien, elokuvien, TV-sarjojen tai visuaalisten tehosteiden luomiseen. Toisin kuin Autodeskin toinen 3D-mallinnus ohjelma 3ds Max, Maya tukee Windowsin lisäksi Linux ja Mac OS X käyttöjärjestelmiä. Mayasta on saatavilla myös opiskelijoille suunnattu Educational Licence ilmaiseksi ohjelman opettelua varten. (Autodesk, 2017.)

Modo on 3D-mallinnus ohjelmisto Windowsille, Linuxille ja Macille. Modo tukee polygonimallinnusta, sculptausta, 3D-maalausta, animaationa ja renderausta. Modon on kehittänyt Foundry. Modon takana on uraauurtavan 3D ohjelmiston LightWave 3D:n luojat. Modoa on käytetty muun muassa elokuva- ja peliteollisuudessa. (Foundry, 2017.)

3.4 Yhteenveto

Blenderin suurimpana valttikorttina verrattuna 3ds Maxiin on hinta. Siinä missä 3ds Max useita tuhansia, Blender on täysin ilmainen. Blender on avoimen lähdekoodin ohjelma ja sitä kehitetään jatkuvasti. Blenderiin löytyy paljon tutoriaaleja, mutta kaupallinen tuki ei ole niin helposti saatavilla kuin 3ds Maxissa. Koska 3ds Max on kaupallinen sovellus, siihen löytyy myös paljon kaupallista tukea ja Autodeskin omia tutoriaaleja.

3ds Max on Blenderiä enemmän ammattimaisten yritysten käytössä. Jos haluaa työllistyä alalle, 3ds Maxin osaaminen täytyy jossain vaiheessa oppia. Blender on yleisessä käytössä indie-pelifirmojen keskuudessa. Koska Blender on ilmainen, sitä voivat käyttää niin harrastelijat kuin ammattilaiset. Blenderiä käyttävät pienemmät ja uudemmat pelistudiot, kun taas 3ds Max on yleisemmin käytössä isommissa pelistudioissa.

3ds Maxin ja Blenderin työkuluissa ei ole suurempia eroavaisuuksia. Blender nojaa 3ds Maxia enemmän pikanäppäinten käyttämiseen, kun taas 3ds maxissa on enemmän valikoita. 3ds Max nojaa enemmän hiiren käyttämisen, kun taas Blenderiä käyttäessä näppäimistö on enemmän käytössä. Blenderin pikanäppäimet voivat tuntua 3ds Maxia käyttäneestä haastavalta ja sitten taas 3ds Maxin valikot voivat tuntua Blenderiä käyttäneestä haastavalta. 3ds Max tukee vain Windows käyttöjärjestelmää, kun taas Blender toimii lähes kaikilla käyttöjärjestelmillä. Molemmista ohjelmistoista mallit saadaan ulos fbx muodossa, joten mallien tuomisessa pelimoottoriin ei ole eroa. Kaikilla ohjelmistoilla saa periaatteessa samanlaisen lopputuloksen, joten on käyttäjästä kiinni, mitä ohjelmistoa loppujen lopuksi haluaa käyttää. Opinnäytetyöhön 3D-mallinnusohjelman valinta osui 3ds Maxiin, sen helppouden ja käyttöominaisuuksien kattavuuden takia.

4 3D-MALLINNUSPROSESSI

3D-mallintamisessa voi käyttää lukuisia eri tapoja ja menetelmiä. Työvaiheet voidaan tehdä eri järjestyksessä tai jättää tarpeen mukaan työvaiheita välistä. Tässä kappaleessa käyn läpi kaikki 3D-mallinnuksen työvaiheet järjestyksessä, jonka olisi tarkoitus olla mahdollisimman optimaalinen ja kustannustehokas.

Lopussa vedän yhteen työvaiheet, joista koostan työtilaajalle kustannustehokkaan ja 3D-mallinnuksen työnkulunmallin. Malli on kaavio, josta käy ilmi työnteen järjestys, käytetyt ohjelmistot ja aika, kuinka kauan kunkin vaiheen toteuttamisessa menee. Tässä 3D-mallinnuksen työnkulunmallissa ohjelmistoina tullaan käyttämään: 3D-mallintamiseen 3ds Maxia, teksturointiin ja tekstuurikarttojen tekemiseen Substance Painteria ja pelimoottorina käytetään Unreal Engine 4:ää.

4.1 Suunnittelu vaihe

Ensimmäinen vaihe tuotettaessa 3D-mallia on suunnittelu. Tarve mallille voi tulla monista eri lähteistä. Seuraavien asioiden miettimistä kannattaa harkita ennen mallintamisen aloittamista:

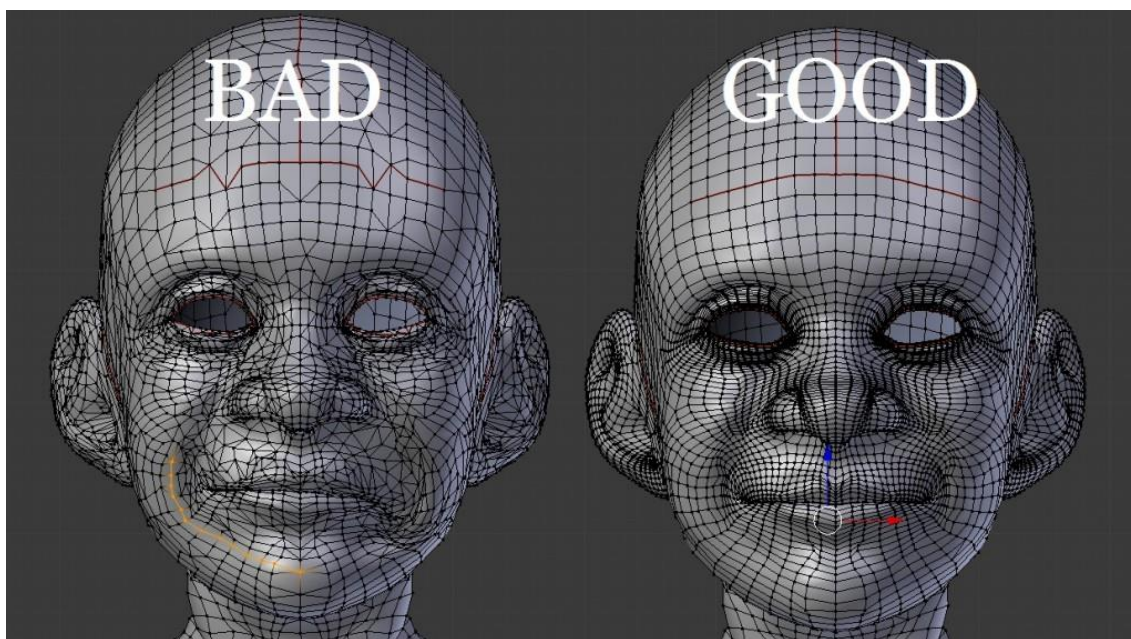
- Kuinka tärkeä mallinnettava asia on pelille?
- Mallinnettavan objektin koko ja mitat?
- Tarvitseeko objektin yksityiskohdat näkyä ja kuinka tarkasti?
 - o Pystyykö yksityiskohdat tekemään materiaalilla?
 - o Tarvitsevatko yksityiskohdat mallintaa?
- Onko kyseessä asia, jota pelaaja näkee usein?
- Onko pelaajalla aikaa tutkia esinettä tarkemmin?
- Onko esine interaktiivinen, tarvitseeko pelaajan olla vuorovaikutuksessa esineen kanssa (esimerkiksi nappula)?
- Jos kyseessä on liikkuva asia, miten se liikkuu?
- Onko kyseessä yksinkertainen vai monimutkainen malli?
- Arvio, kuinka paljon mallin tekeminen tulee viemään aikaa
- Tuleeko mallista kuulua ääntä? Jos tulee niin mitä ääntä?

Jos asia jota haluaa alkaa mallintamaan ei ole pelin kannalta tärkeä, sitä on turha mallintaa liian tarkasti. Liian tarkasti mallintaminen vain kuluttaa aikaa ja resursseja, joita voisi käyttää tärkeämpien objektien luomiseen. Kun tarvittava malli on valittu, ennen mallintamisen aloittamista on hyvä käyttää apuna referenssikuvia. Esimerkiksi Googlen kuvahaku on erinomainen paikka aloittaa sopivan referenssikuvan hakeminen. Kannattaa myös pitää mielessä pelin graafinen tyyli. (Pettit 2015.)

4.2 Mallinnus

Mallintamisen voi aloittaa joko tuomalla referenssikuvia mallinnusohjelmaan tai ihan vaan vapaasti mallintamalla ilman referenssi kuvia. Mallinnusvaiheessa myös määritellään, mihin kohtaan mallia tulee materiaaleille paikat.

Mallintaessa kannattaa kiinnittää huomiota mallin topologiaan. Sana topologia viittaa 3D-objektin geometriseen pintaan. 3D-mallia mallintaessa tulee pyrkiä mahdollisimman puhtaaseen topologiaan. Tämä tarkoittaa sitä, että mallilla on tehokas polygonijakauma, reunapolygonit on sijoitettu asianmukaisesti, mallissa ei ole tai on vähän kolmikulmaisia polygoneja, sekä mallissa ei ole venyneitä tai vääristyneitä kohtia. Tämä säästää aikaa ja takaa sen, että optimointivaiheessa on vähemmän työtä. Kun malli viedään pelimoottoriin, huono topologia syö turhaan tehoja ja on suoristuskäytön kannalta ongelmallinen. (Slick 2016.)

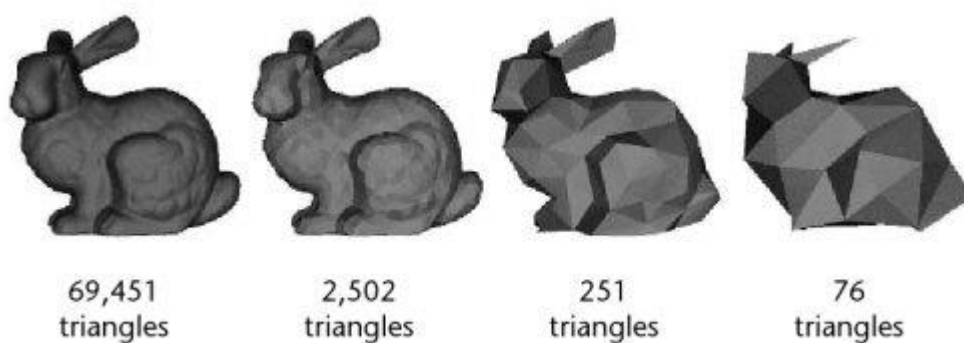


Kuva 15. Esimerkki hyvästä ja huonosta kasvojen topologiasta. (Thilakanathan Studios, 2016.)

4.3 Optimointi ja viimeistelyvaihe

Kun 3D-malli on valmistunut, optimointivaiheessa mallista poistetaan kaikki ylimääräinen. Tällä prosessilla taataan se, että kun mallin tuodaan pelimoottoriin, se ei ole liian raskas pelimoottorille. Tässä vaiheessa, jos objekti on staattinen, poistetaan kaikki ne polygonit, joita pelaaja ei näe. Lisäksi mallia käydään läpi ja poistetaan ylimääräiset edget ja verteksit, kuitenkin niin että malli säilyisi ulkoisesti ennallaan. Sen lisäksi, jos mallissa on reikiä tai muita vääristymiä, ne korjataan.

Peleissä voidaan käyttää myös LOD-menetelmää. LOD (Level of Detail) eli yksityiskohtien taso tarkoittaa, että yhdelle objektille määritellään useita eri versioita mallista laskien aina polygonimäärää. Mitä lähempänä objektia pelaaja on, se versio mallista jossa on eniten polygoneja, näytetään. Kun taas pelaaja liikkuu kauemmaksi objektista, objekti vaihtuu versioon, jossa on vähemmän polygoneja ja niin edelleen. (Silverman 2013.) LOD-mallit voidaan joko luoda manuaalisesti mallinnusohjelmassa tai käyttämällä ulkoista tarkoitukseen soveltuvaa ohjelmistoa. Useimmissa pelimoottoreissa on kuitenkin LOD-asetukset, jolloin ylimääräisiä LOD-malleja ei välttämättä tarvita.



Kuva 16. Esimerkkejä yhden objektin LOD-malleista (Polygon Reducer, 2008.)

Optimointivaiheessa tulee myös huomioida mallin skaalaus. Kun mallin on jo mallinnusohjelman puolella oikean kokoinen, mallin tuominen pelimaailmaan nopeutuu, koska mallin kokoa ei tarvitse enää säätää ja kaikki mallit ovat samankokoisia.

Mallintaessa hyvä nyrkkisääntö onkin, että mallit ovat suurin piirtein samankokoisia kuin oikeassa elämässä.

4.4 Riggaus ja animointi

Jos mallinnetussa mallissa on liikkuvia osia, tai sen tarvitsee liikkua jollakin tietyllä tavalla, se täytyy sen rigata ja sen jälkeen animoida. Sana "Riggaus" tarkoittaa suurin piirtein suomennettuna "köysitystä". Riggaus tarkoittaa digitaalisen luurankon luomista 3D-malliin. Rigattu malli koostuu nivelistä ja luista, joista jokainen toimii ikään kuin kahvana, jota taivutteleamalla malli saadaan haluttuun asentoon.

Riggauksen jälkeen ennen animointia, malli pitää skinnata. Skinnaus (englanniksi skinning) tarkoittaa prosessia, jolla sidotaan rigatut luut mallin meshiin. Prosessin avulla määritetään kuinka paljon mikäkin kohta meshistä, liikkuu mitäkin luuta liikuttaessa.

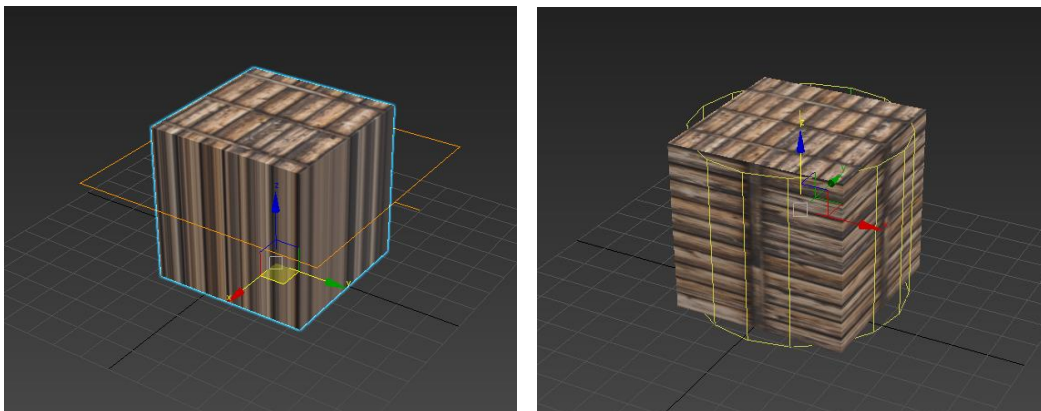
Animoinnissa riggauksessa luotuja luita liikutellaan halutulla tavalla. Luita voidaan joko liikuttaa yksitellen tai useampia samaan aikaan. Animointi toimii 3ds Maxissa Timeline toiminnon avulla. Timeline toiminnolla luita liikutellaan yksi frame kerrallaan, jonka jälkeen frameet merkataan Autokey -toiminnolla.

4.5 UV-mappaus

Kun tarvittava esine on mallinnettu, se täytyy UV-mapata. UV-mappaus eli UV-kartoitus tarkoittaa 3D-mallin pintojen tasoittamista 2D-pinnoiksi, jotta tekstuurikarttoja voidaan liittää malliin. Kirjaimet U ja V merkitsevät 2D-tekstuurin akseleita, koska kirjaimia X, Y ja Z käytetään 3D-mallin akseleiden merkitsemiseen. Ennen UV-mappauksen aloittamista mallin tulisi olla valmis, sillä jos malliin tulee UV-mappauksen jälkeen muutoksia, koko UV-mappaus prosessi on aloitettava alusta. Monimutkaisten objektien kohdalla UV-mappauksen nopeuttamiseksi, UV-mappauksen voi tehdä käyttämällä muita sovelluksia kuin mallinnusohjelmistoa.

UV-mappaus tyylejä on useita. 3ds Maxissa UV-mappaukselle on omia määrittäjä, perinteinen UVW Map ja hieman kehittyneempi Unwrap UVW. Perinteisiä kartoitusmodifikaattoreita ovat muun muassa tasomainen, lieriömainen, pallomainen, kutistus, laatikkomainen, Pinta ja XYZ to UVW. (Autodesk 2017.)

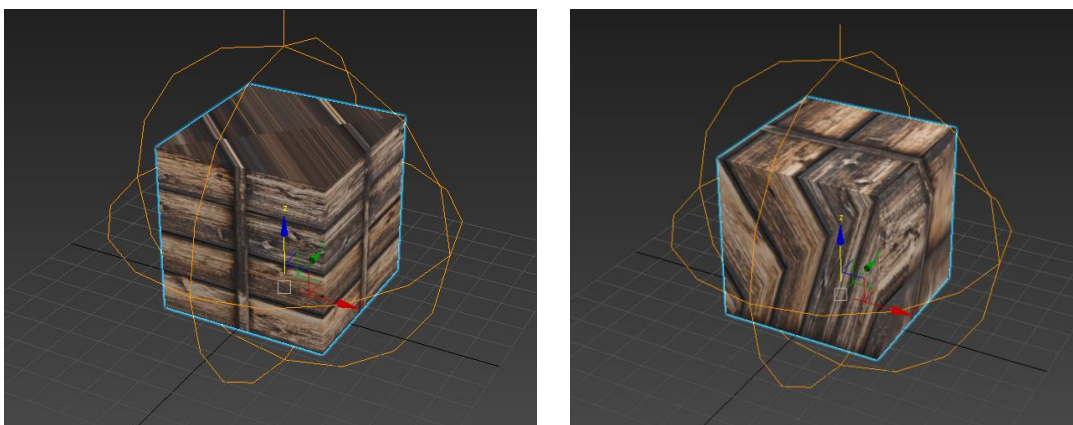
Tasomainen mappaus eli planar-mappaus on kaikkein yksinkertaisin kartoitusmodifikaattoreista. Planar-mappaus heijastaa UV:t tason (plane) läpi objektiin. Planar mappaus sopii esineille, jotka ovat tasaisia, kuten seinät tai lattiat, tai objekteille, joista pelaajalla näkyy vain yksi puoli. Planar mappaus ei sovellu monimutkaisille objekteille, koska se yleensä venyttää polygoneja, jotka eivät kohdistu heijastettuun mappiin suoraan. (Banninga, 2004.)



Kuva 17. Taso- ja sylinterimappaukset

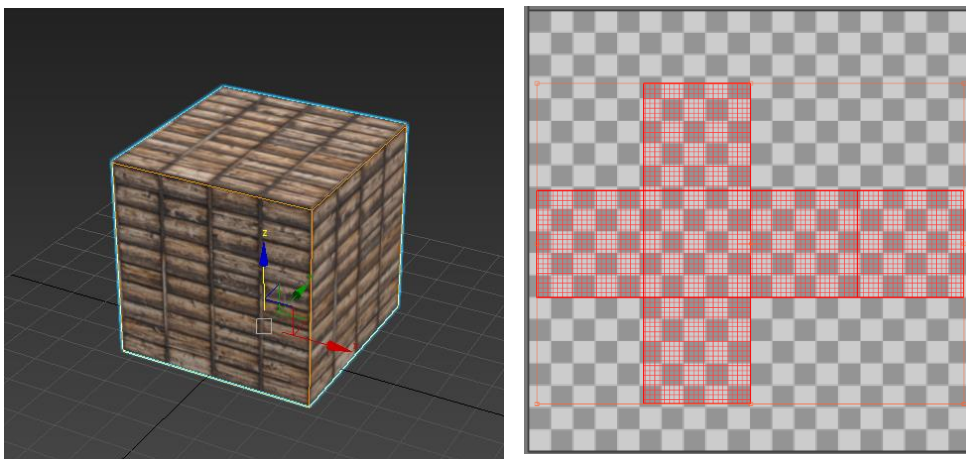
Sylinterimappaus luo sylinterimäisen muodon mallin ympärille. Tämä tekniikka sopii parhaiten muotoihin, jotka näkyvät kokonaan sylinterin muodossa, ilman ulkonevia tai onttoja osia. Tämä tekniikka luo saumat kohtaan, jossa sylinterin reunat kohtaavat. (Autodesk 2017.)

Spherical mappaus eli pallomainenmappaus ympäröi objektin heijastaen mapin pallon muotoon. Saumat tulevat pallon ylä- ja alareunaan, eli niin sanotuille navoille. Pallomainen kartoitus sopii esineille, jotka ovat suurin piirtein pallon muotoisia, ilman ulkonevia tai unttoja kohtia. Shrink Wrap tai kutistusmappaus käyttää pallomaista kartoitusta, mutta katkaisten kartan kulmat ja liittäen ne yhteen pisteeseen. Shrink Wrap -mappaus on hyödyllinen, jos halutaan piilottaa kartoituksen singulaarisuus. (Banninga, 2004.)



Kuva 18. Pallo- ja kutistusmappaukset.

Laatikkomaisessa mappauksessa eli box mappauksessa kaikki kuution kuusi sivua toimivat samoin tavoin kuin planar mappaus. (Autodesk.) Box mappaus sopii teknisille ja arkkitehtonisille objekteille. Box mappaus ei toimi orgaanisille muodoille, tai muodoille joissa tarvitaan tarkkaa mappautusta. Face mappauksessa kartta kopioidaan objektin jokaisen polygonin pintaan. (Banninga, 2004.)



Kuva 19. Laatikkomappaus ja laatikon unwrapattu uv-kartta.

Unwrappaus tarkoittaa prosessia, jossa luodaan objektille erityiset ja objektille varta vasten tehdyt UV-mapit. Unwrappauksessa meshin pinta avataan ja tasoitetaan niin, että 3D-mallille pystytään tekemään 2D-tekstuuri, joka sopii 3D-objektille. Kun objekti on unwrapattu, sitä pystyy muokkaamaan ja järjestelemään suorituskyvyn kannalta optimaalisempiin asentoihin. Unwrappaus on tärkeää, jos tekstuurin täytyy kuvata objektia tarkasti. UV-mappaus johtaa korrelaatioon kuvasta ja siitä, miten se näkyy tekstuurina, kun se on kartoitettu 3D-meshin pintaan.

4.6 Beikkaus

Tekstuurien beikkaus (englanniksi baking) tarkoittaa prosessia, jossa toisesta mallista siirretään yksityiskohtia toiseen malliin, useimmiten korkeapolygonisesta mallista matalapolygoniseen malliin. Beikkaustyökalu aloittaa tietyn etäisyyden päästä mallista, ja alkaa heittää säteitä sisäänpäin kohti toista mallia. Kun säde leikkaa toisen mallin kanssa se rekisteröi pinnan yksityiskohtia ja tallentaa tiedon sen tekstuurikarttaan, käyttäen ensimmäisen mallin tekstuurikoordinaatteja. Beikkausta tyypillisesti käytetään generoitaessa light mappeja, normal mappeja tai LOD-malleja. Beikatessa matalapolygonisten ja korkeapolygonisten mallien tulee olla täysin päällekkäin ja täysin

samankokoisia. Ennen beikkausprosessin aloittamista matalapolygonisen mallin uv-kartoituksen tulee olla valmis.

4.7 Exporttaus

Exporttaus (englanniksi exporting) tarkoittaa 3D-ohjelmalla tehdyn valmiin mallin viemistä sellaiseen muotoon, että sen voi avata muissa ohjelmissa. Kun malli on valmis, se tallennetaan useimmiten fbx tai obj muotoon. 3ds Maxissa voi exportata koko scenen kerralla, tai vain valittuja yksittäisiä objekteja käyttämällä Export Selected -toimintoa. Monimutkaiset objektit voidaan myös exportata monessa eri osassa, näin helpottaen UV unwrappausta ja materiaalien laatua.

Jos objektissa on käytetty tasoitusryhmiä tai objektissa on käytetty TurboSmooth -toimintoa, exportausvaiheessa nämä voidaan, joko exportata suoraan objektiin, tai jättää pois exportatusta mallista. Exportatessa animaatioita tulee käyttää fbx tiedostomuotoa. Staattisiin objekteihin voidaan käyttää muita tiedostomuotoja. 3ds Maxista saadaan exportattua objektit esimerkiksi suoraan Adobe Illustrator –ohjelmaan.

4.8 Teksturointi

Teksturoinnin ja materiaalin luomisen tarkoituksena on elävöittää mallinnettua 3D-mallia. Tekstuuri on kuva, joka on tuotu suoraan Photoshopista tai vastaavasta ohjelmasta. Materiaali taas on yksityiskohdat siitä, miltä kohteen tulisi näyttää. Materiaali koostuu monista tekstuureista, jotka määrittelevät materiaalin yksityiskohdat. Yksi tekstuuri voi määritellä materiaalin värin, toinen voi toimia normalmappina ja kolmas voi määrittää kohteen läpinäkyvyyden. Materiaalin voi myös luoda pelimoottorin puolella syöttämällä manuaalisesti parametrejä.

Ennen valmiin 3D-mallin tuomista pelimoottoriin se täytyy teksturoida. Teksturointiin ja materiaalien tekemiseen löytyy useita eri ohjelmistoja. Tekstuurien tekemisessä voi käyttää yksinkertaisimmillaan Photoshopia tai jotain vielä yksinkertaisempaa ohjelmaa. Materiaalien tekemiseen tarvitsee toisenlaisen ohjelmiston. Materiaalien tekemisessä käytetään apuna Allegorithmicin kehittämää Substance Painteria. Substance Painterilla ei itsessään tehdä materiaaleja, mutta teksturointiin se sopii erinomaisesti. Jos

materiaalin haluaa tehdä alusta loppuun, käytetään Substance Designeria. Substance Painterissa on paljon valmiita materiaaleja ja Designeristä saa materiaalit suoraan Painteriin.

Tärkeimpiä asioita, mitä teksturoidessa kannattaa miettiä:

- Mistä materiaalista esine on tehty? Puusta, muovista, metallista, betonista, kankaasta, nahasta?
- Onko esineen värityksellä pelin kannalta väliä?
- Tarvitseeko esineen heijastaa valoa? Ja kuinka paljon?
- Näkyykö esineestä läpi? Ja kuinka paljon?
- Onko mallin tekstuuri animoitu tai liikkuva?
- Mikä on pelin graafinen tyyli?
- Tarvitseeko esine jotain tarkempia yksityiskohtia?
- Onko esine tärkeä pelille?
 - o Jos esine on pieni ja epäolennainen pelille, esineen karttojen ei tarvitse olla kovinkaan tarkkoja.
 - o Jos esine on tärkeä, karttojen on hyvä olla mahdollisimman tarkkoja, mutta kuitenkin ottaen huomioon suorituskäytön.

Jos mallin tekstuurit heijastavat liikaa tai värit eivät sovi, niin usein näitä asetuksia pystyy vielä pelimoottorin puolella säätämään. Yksinkertaisimmillaan materiaaliin tarvitsee vain basecolor-kartan. Tämä ei kuitenkaan ole suositeltavaa, koska mallin graafinen taso kärsii. Esineen elävöittämiseksi materiaalissa tulee minimissään olla basecolor-, metallic- ja roughness- mapit. Kaikki nämä mapit saa suoraan exportattua Substance Designeristä/Painterista.

Basecolor map on tärkein tekstuuri. Se määrittelee, millainen on objektin perusväritys. Normal mappia käytetään yleisesti saamaan low poly mappeihin high poly mappien yksityiskohdat. Kartan jokainen pikseli tallentaa jokaisen pinnan kaltevuus pisteen alkuperäisestä korkearesoluutioisesta meshistä. Tämä luo illuusion pinnan tarkemmista yksityiskohdista, kaarevuudesta, korkeudesta ja syvyydestä. Sivustapäin katsottuna objektin siluetti ei kuitenkaan muutu. Normal mappeja saa luotua joko beikkaamalla objekti tai maalamalla objektin päälle käyttämällä teksturointiohjelmia. Yleensä normal mapit ovat sävyiltänsä vaalean sinisestä violettiin. (Slick 2016)

Unreal Engine 4 käyttää Roughness ja Metallic mappeja, Specular ja Glossiness mappien sijaan muista pelimoottoreista poiketen. Roughness ja metallic mapit

määrittävät kuinka paljon objektin materiaali heijastaa valoa. Jos roughness mapin parametrin arvo on 1 ja metallic mapin 0, materiaali ei heijasta valoa. Jos taas parametrien arvot ovat roughness 0 ja metallic 1, materiaali on peilimäinen, eli se heijastaa lähes kaiken valon.

4.9 Importtaus

Importtaus (englanniksi importing) tarkoittaa valmiin 3D-mallin ja tehtyjen tekstuurien tuomista pelimoottoriin. Valmiista tekstuurikartoista tehdään pelimoottorin puolella materiaali. Prosessia nopeuttaa, jos kaikki tekstuurit ovat jo valmiina eikä niitä tai mallia tarvitse enää muokata. Mallin pelimoottoriin tuomiseen jälkeen materiaalit, joita ei tehty teksturointi osassa (esimerkiksi lasi, peili tai yksiväriset materiaalit) tehdään pelimoottorissa.

5 MALLINNUSPROSESSIN HYÖDYNTÄMINEN PELIPROJEKTIIN

Tässä kappaleessa tehdään 3D-mallin peliin alusta loppuun käyttäen edellä läpikäytyjä 3D-mallinnusprosessin eri vaiheita. 3D-mallin tekemisessä on käytetty 3ds Max ja Substance Painter -ohjelmistoja. Valmis malli viedään pelimoottoriin, jossa se viimeistellään vastaamaan pelin tarpeita.

5.1 Lyhyesti peliprojektista

Opinnäytetyö toteutetaan osana Morrow Gamesin peliprojektia. Peliä kehitetään PC-alustalle ja pelimoottorina käytetään Epic Gamesin kehittämää Unreal Engine 4:sta. Pelin päähahmo on harrastelijakuvaaja, jonka tarkoituksena on kuvata hylättyä rakennusta. Pelin ohjelmoinnissa käytetään pelkästään Unreal Enginen blueprinttejä.

Unreal Engine 4 käyttää Blueprints Visual Scripting -nimistä järjestelmää, joka mahdollistaa pelien kehittämisen käyttäen node-pohjaista käyttöliittymää, manuaalisesti koodaamisen sijaan. Tämä nopeuttaa ja helpottaa pelien kehittämistä ja se on joustava ja tehokas esimerkiksi suunnittelijoille tai artisteille, sillä se antaa mahdollisuuden käyttää lähes kaikki käsitteitä ja työkaluja, jotka ovat yleensä vain ohjelmoijille käytettävissä. (Epic Games, 2017.)

5.2 Suunnitteluvaihe

Suunnitteluvaiheessa malliksi valikoitui videokamera. Mallia lähdettiin toteuttamaan, koska peliin tarvittiin videokamera. Peli sijoittuu hylättyyn rakennukseen, joten lähes kaikki pelin materiaalit ovat hieman ränsistyneitä ja rähjäisiä. Pelin päähahmo kuitenkin tuo kameran mukanaan, joten kameran ei tarvitse olla liian kuluneen näköinen. Kameran tulisi kuitenkin olla käytetyn näköinen.

Referenssikuvien etsintä aloitettiin ihan käyttämällä perinteistä Googlen hakutoimintoa ja etsimällä erilaisia videokameramalleja. Videokameramalliksi valikoitui Sony Handycam CX675, jonka pohjalta päädyttiin mallintamaan peliin tarvittava videokamera. Sonyn videokamera valikoitui siksi, että se oli juuri sen näköinen, jota

peliiin tarvittiin sekä siitä löytyi runsaasti ja useista eri kulmista referenssikuvia. Alun perin kameran näyttöön oli tarkoitus tulla reaaliaikainen renderöinti kameran linssistä, mutta tästä luovuttiin ja päädyttiin vain animoituun tekstuuriin. Reaaliaikainen renderöinti hylättiin sen työläyden ja suorituskyvyn takia.



Kuva 20. Videokameran referenssikuvat (Sony.)

5.3 Mallinnusvaihe

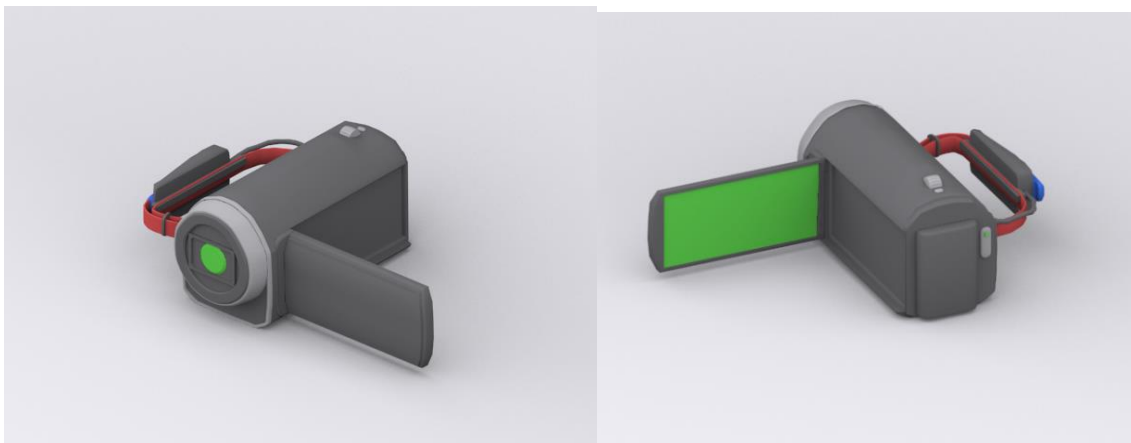
Mallin työstäminen aloitettiin tuomalla referenssikuvat 3ds Maxiin, jonka jälkeen luotiin laatikko, joka toimi kameran runkona ja jota verteksejä ja edgejä lisäämällä muokattiin pikkuhiljaa referenssikuvia vastaamaan. Mallinnus aloitettiin itse videokameran rungosta sen ollessa tärkein osa ja pohja koko mallille. Pienemmät osat luotiin kameran rungon muotojen ollessa lähes valmiit.

Pelaajan tulee pelissä tarkastella lähemmin mallia, joten videokamera mallinnettiin mahdollisimman tarkasti, kuitenkin pitäen mallin suorituskyvyn mielessä. Koska pelaajan tulee tarkastella mallia lähemmin, videokameran tekstuurien resoluutio tulee olla myös sen verran korkeatasoinen, että pelaaja ei kiinnitä resoluutioon sen enempää huomiota. Kameran pintojen kaarevuus mallinnettiin käsin ja käyttämällä smoothing groupeja, TurboSmoothin tai NURBSin sijasta.

Videokameran kahvaosa tehtiin myös referenssi kuvien avulla, jonka jälkeen niiden päälle lähdettiin spline-työkalun avulla tekemään kameran kahvan muotojen mukaisia linjoja. Tämän jälkeen splinet muutettiin joko nelikulmion, tai sylinterin muotoiseksi saaden niille haluttua syvyyttä ja muotoja.

Videokameran näytön mallinnus aloitettiin kameran rungon tapaan laatikosta, jonka jälkeen sitä muovattiin haluttuun muotoon lisäämällä edgejä connect -työkalun avulla. Itse näyttöosaa extrudattiin hieman sisäänpäin, koska haluttiin korostaa näyttöä.

Malliin ei tullut niin suurta tai monimutkaista animaatiota, että se olisi tarvinnut mallinnusohjelman puolella suurempaa animointia, joten animointivaihe 3ds Maxin puolella jätettiin pois ja animointi suoritettiin pelimoottorin puolella myöhemmin mallin ollessa valmis.



Kuva 21. Valmis 3D-malli ilman tekstuureja ja valmiiksi määritellyillä materiaalipaikoilla.

5.4 Optimointi

Valmiin mallin optimointiin ei kulunut kauaa aikaa. Mallista poistettiin ylimääräisiä edgejä ja kaikki päällekkäiset verteksit hitsattiin yhteen. 3D-mallin LODit tehtiin pelimoottorin puolella, käyttäen Unreal Enginen omia ja valmiiksi generoituja LOD-asetuksia. Mallin optimointi oli tärkeää, koska kyseessä on pelin kannalta tärkeä malli, ja jonka kanssa pelaaja joutuu olemaan vuorovaikutuksessa. Koska mallia tulee tarkastella lähemmin se ei saa olla liian optimoitu.

5.5 UV Mappaus

Mallin UV-mappaus toteutettiin Unwrap UV -työkalulla ja käyttäen Flatten Mapping-toimintoa. Mallia unwrapatessa todettiin, että perusasetuksilla käyttämällä sai ihan riittävän hyvän tuloksen, joten mallin UV-mappeja tarvitsi vain hiukan optimoida ja mallin osasia siirrellä siten että uv mappi olisi mahdollisimman tarkka ja pelin kannalta hyvin optimoitu. Kameran kahva, näyttö ja itse kamera osa unwrapattiin erikseen, koska mallin UV-mappien haluttiin olevan mahdollisimman tarkat.

5.6 Exporttaus

Mallin liikkuvien osien, UV-mappauksen ja materiaalipaikkojen helpottamiseksi malli exportattiin kolmessa osassa: näyttö, kahva ja itse kamera. Näyttö exportattiin erikseen, koska näytön avaaminen ja kiinnimeneminen, on tarkoitus animoida vasta pelimoottorin puolella. Mallin tasoitusrhyhmät exportattiin myös mallin kanssa. Valot, kamerat, tekstuurit ja materiaalit jätettiin exporttaamatta niiden ollessa tarpeettomia. Tällä taattiin se että, exportatussa tiedostossa ei ollut mitään ylimääräistä.

5.7 Teksturointi

Mallin teksturointi toteutettiin käyttämällä Substance Painteria. Mallin päämateriaalit oltiin tehty, joko ennen projektia Substance Designerissä, tai materiaaleina käytettiin jo valmiina olevia materiaaleja.

Mallin tarkemmat yksityiskohdat, kuten tekstit ja nappulat joita ei mallinnettu tehtiin Photoshopilla, jonka jälkeen ne yhdistettiin mallin tekstuureihin Painterin puolella. Malliin tuli 5 materiaalipaikkaa. Materiaaleiksi tuli muovi, metalli, nahka, kangas ja näyttö.

Koska pelin graafinen tyyli on mahdollisimman realistinen niin, materiaalit olivat mallin kannalta suuressa osassa. Teksturoinnissa käytettiin mahdollisimman realistisia materiaaleja. Itse kameran materiaaliksi haluttiin vähän kulunut ja likainen pinta. Kameran materiaalin tulisi olla sen näköinen, että kamera on ollut hyvässä käytössä pelin päähahmon toimesta tai se on ostettu käytettynä. Kameran materiaaliksi valikoitui musta, mattapintainen ja kulunut muovi.

Kameran näytön materiaalin luominen toteutettiin vasta pelimoottorin puolella, koska näyttöön tulee liikkuva materiaali simuloimaan oikeata näyttöä. Lisäksi halutaan että, näyttö toimii myös valon lähteenä. Kameran metalliosiksi tulee hiukan kulunut alumiini. Vaikka kyseessä oli metalli, materiaalin ei haluttu heijastavan liikaa valoa, joten tekstuuri tehtiin lähes mattapintaiseksi. Kameran hihnan materiaaleiksi tuli nahka ja kangas. Molemmat näistä ovat matalaresoluutioisia, sillä materiaalit eivät näy hihnan ollessa tumma ja vähemmän esillä kuin kameran muovi ja metalliosat.

Kameran kaikki tarkemmat yksityiskohdat, joita ei mallinnettu tehtiin Photoshopissa, jonka jälkeen ne yksitellen maalattiin Substance Painterissa 3D-mallin pintaan. Photoshopissa tehtiin myös kaikki kameran tekstit ja tarrat sekä kameran kuvitteellisen valmistajan logo.

5.8 Mallin vienti pelimoottoriin

Koska videokameran näytön tarvitsee avautua, näytön liikkuminen animoidaan pelimoottorin puolella. Pelimoottorin puolella animoiminen ei ole niin tarkkaa kuin mallinnusohjelmassa, mutta koska kyseessä on hyvin yksinkertainen ja vain yhteen suuntaan liikkuminen, animointi on helpompi tehdä pelimoottorin puolella. Pelimoottorin puolella mallin osaa liikutellaan, eikä varsinaisesti animoida, joten näin ollen malliin ei tarvinnut rigata tai luoda luita. Pelimoottorissa huomattiin, että mallin lightmap resoluutio oli hyvä, sillä UV-mapeista ei heijastunut outoja varjoja objektiin.

Kameran näytön tekstuurit toteutettiin käyttämällä "panner" -toimintoa, joka liikuttaa kuvaa joko vaaka-, pysty- tai molemmissa tasoissa yhtä aikaa. Ensin luotiin sopiva kuva Photoshopissa, jonka jälkeen sitä liikuteltiin halutulla tavalla saaden valkoista kohinaa kuvaavan tekstuurin.

Kameran näytön aukeaminen toteutettiin käyttäen Unreal Enginen Blueprinttejä. Kameran näytöstä luotiin blueprintti, jossa näytön kääntymisen kulma ja nopeus määritettiin Timeline- toiminnon avulla ja käyttämällä SetRelativeRotation -nodea. Kameran näyttö aukeaa, kun pelaaja on vuorovaikutuksessa kameran kanssa. Lisäksi näyttöön kiinnitettiin himmeä lamppu, johon luotiin välkkyvä efekti tuomaan lisää realismia kameraan.



Kuva 22. Valmis videokameran 3D-malli teksturointuna ja pelimoottoriin vietynä.

6 YHTEENVETO

Opinnäytetyön tavoitteena oli luoda asiakkaalle 3D-mallinnuksen työnkulun malli ja sen perusteella mallintaa esimerkki 3D-mallista, jonka oli tarkoitus tulla osaksi asiakkaan projektia. Lisäksi opinnäytetyön tavoitteena oli tutkia tarkemmin 3D-mallinnukseen liittyviä työvaiheita tarkemmin ja näin saada itselle parempi käsitys työvaiheista tulevaisuutta ajatellen.

Opinnäytetyön tavoitteet saavutettiin, sillä asiakas oli tyytyväinen sekä työnkulun malliin, että sen pohjalta tehtyyn 3D-malliin. Esimerkki 3D-mallina toiminut kannettava videokamera onnistui siinä määrin, että sitä voidaan käyttää siinä projektissa mihin se oli alun perin suunniteltu tulevankin. Myös 3D-mallinnuksen työnkulun mallia voidaan käyttää uusien työntekijöiden tai työharjoittelijoiden kouluttamisessa.

Peliprojektia, jonka osana opinnäytetyö suoritettiin, on tarkoitus Morrow Gamesin toimesta kehittää jatkossa joskus tulevaisuudessa. Projekti ei ole ainakaan tällä kovin tärkeä yritykselle, vaan sitä kehitetään eteenpäin muiden töiden ohella.

LÄHTEET

- Autodesk 2015. Introduction to polygons. Viitattu 8.7.2017. <https://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Polygons-overview-Introduction-to-polygons-htm.html>.
- Autodesk 2016. Smoothing Groups. Viitattu 6.7.2017. <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-1244162D-A063-486C-BD9B-168466F6488B-htm.html>.
- Autodesk 2014. UVW Map Modifier. Viitattu 22.5.2017. <https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-78327298-4741-470C-848D-4C3618B18FCA-htm.html>.
- Banninga, R. 2004. UV Mapping Tips and Tricks. Viitattu 22.5.2017. http://www.gamasutra.com/view/feature/130484/uv_mapping_tips_and_tricks.php.
- Blender Foundation 2017b. About. Viitattu 19.5.2017 <https://www.blender.org/about/>.
- Blender Foundation 2017a. Blender History. Viitattu 9.5.2017 <https://www.blender.org/foundation/history/>.
- CGMeetup 2015. Smaug Speed Modeling in Zbrush. Viitattu 15.8.2017. <http://www.cgmeetup.net/home/smaug-speed-modeling-in-zbrush/>.
- Holm, P. 2005 3ds Max history. Viitattu 9.5.2017 <https://www.3dbuzz.com/forum/threads/97960-3ds-max-history>.
- Maxon. Cinema 4D in the movies. Viitattu 22.5.2017. <https://www.maxon.net/en-us/industries/movies-vfx/>.
- Papageorgiou, M. 2017. Top 7 of the best Sculpting software for 3D modeling. Viitattu 1.10.2017. <https://www.sculpteo.com/blog/2017/09/20/top-7-of-the-best-sculpting-software-for-3d-modeling>.
- Pettit, N. 2015 Asset Workflow for Game Art: 3D modeling. Viitattu 16.5.2017. <http://blog.teamtreehouse.com/asset-workflow-game-art-3d-modeling>.
- Pluralsight 2014. Understanding Skinning – The Vital Step for Any Rigging Project. Viitattu 19.5.2017 <https://www.pluralsight.com/blog/film-games/understanding-skinning-vital-step-rigging-project>.
- Polygon Reducer 2008. Reducing level of detail. Viitattu 16.8.2017. <http://polygon-reducer.pc-guru.cz/reducing-level-of-detail>.
- Rhinoceros 2017. What are NURBS? Viitattu 7.8.2017. <https://www.rhino3d.com/nurbs>.
- Silverman, D. 2013. 3D Primer for Game Developers: An Overview of 3D Modeling in Games. Viitattu 28.5.2017. <https://gamedevelopment.tutsplus.com/articles/3d-primer-for-game-developers-an-overview-of-3d-modeling-in-games--gamedev-5704>.
- Slick, J. 2016a. The many, linked definitions of “Spline”. Viitattu 15.5.2017. <https://www.lifewire.com/definitions-of-spline-2177>.

Slick, J. 2016b. Topology in 3D animation. Viitattu 19.5.2017. <https://www.lifewire.com/topology-in-3d-animation-2181>.

Slick, J. 2016c. Surfacing 101 – The Basics of Texture Mapping. Viitattu 5.7.2017. <https://www.lifewire.com/texture-mapping-1956>.

Thilakanathan Studios 2016. Why do we need topology in 3D modeling? Viitattu 15.8.2017. <http://thilakanathanstudios.com/2016/09/why-do-we-need-topology-in-3d-modeling/>.

Time & Trends Academy 2016. History of Autodesk 3ds Max. Viitattu 9.5.2017 <http://timesandtrendsacademy.com/history-autodesk-3ds-max/>.

Unreal Engine 4. Blueprints Visual Scripting. Viitattu 14.8.2017. <https://docs.unrealengine.com/latest/INT/Engine/Blueprints/>.

WiseGEEK 2017. What is 3D modelling? Viitattu 9.5.2017 <http://www.wisegeek.com/what-is-3d-modeling.htm>